

C/E网系统中交叠式序列的计数

吴振寰 高莹 吴哲辉

(山东科技大学信息科学与工程学院 青岛 266510)

摘要 在CCS中,Milner把系统中的两个事件处于并发定义为“它们可以按任意顺序发生”。这个关于并发的定义被称之为“交叠式并发”。然而在网论中,Petri把并发定义为“非序”,并指出并发是冲突的精确对偶。Petri关于并发的定义被公认为“真并发”。为了研究这两种并发概念的联系和差异,以C/E系统为模型,分别从这两种不同定义的视角出发,讨论了并发的表现和实质,并对网系统中存在的不同并发组合情况给出了求交叠式序列(根据交叠式并发概念而得出)个数的一组计算公式。

关键词 并发,真并发,交叠式并发,C/E系统,交叠式序列

中图法分类号 TP301 **文献标识码** A

Counting Interlacing Sequence for C/E Systems

WU Zhen-huan GAO Ying WU Zhe-hui

(College of Information Science and Engineering, Shandong University of Science and Technology, Qingdao 266510, China)

Abstract That a pair of events in a system is said to be in concurrency is defined as “they can occur in any order” by R. Milner in CCS (Calculus of Communication System). This definition about concurrency is named as “interlacing concurrency”. However, concurrency is defined as “disorder” by C. A. Petri in net theory. It is recognized as “true concurrency”. In order to investigate the relationship and the difference between these two concepts, we used C/E system as models to discuss the appearance and essentiality of concurrency under both of these two definitions. As a result, a set of formulas to count the number of interlacing sequence (under the concept of interlacing concurrency) upon variable situations were given.

Keywords Concurrency, True concurrency, Interlacing concurrency, C/E system, Interlacing sequence

1 引言

英国科学家罗宾·米尔纳(Robin Milner)由于在开发原语言(metalanguage, ML)和创建通信演算系统(Calculus of Communication System, CCS)等方面的杰出贡献,获得了1991年的图灵奖^[1]。在CCS中,Milner把两个事件并发定义为“它们可以按任意顺序发生”^[2,3]。基于这种理解,CCS成功地解释了用于书写通信协议规约的国际标准语言Lotos,而后者是描述面向对象需求定义中的抽象数据类型和进程的一种有效工具^[1]。

Milner的并发定义实质上是在特定的应用范围内对并发事件的一种处理方式,或者说,是通过序列化处理的各種可能性来解释并发。Milner本人也认为Petri网中定义的并发才是“真并发”。CCS中定义的并发被称之为“交叠式并发(interlacing concurrency)”^[1]。

Petri网模型是德国科学家Carl Adam Petri在他的博士论文中提出来的^[4]。在为文献[5]写的序言中,C. A. Petri指出,1960年的计算机科学由于缺少并发概念,不适于描述物理系统。Petri网模型的提出,是想用一种兼容物理和计算机

科学的概念构架来形式地描述制约通信进程的所有“自然法则”,用并发的概念来丰富计算机科学。关于并发的概念和基本理论,C. A. Petri教授专门有精湛和深刻的论述,并且提出了一组并发公理^[6,8,9]。简要地说,并发就是“非序”,并发是选择(冲突)的精确对偶。

为了探讨“交叠式并发”与“真并发”两个概念之间的差异和联系,本文以C/E系统为对象,对两种并发概念进行比较,并给出在这种网系统中,交叠式序列数目的计算公式。

2 C/E系统中“真并发”与“交叠式并发”概念的比较

基本网系统是用来描述条件和事件的关系的一种网系统。概括地说,基本网系统是权函数和容量函数都恒等于1的库所/变迁系统。条件事件系统则是一种满足特定条件的基本网系统。许多文献都对基本网系统和条件事件系统进行过讨论^[7-9]。下面先对与本文相关的基本概念、术语和记号作一一简述。

三元组 $N=(S, T; F)$ 称为一个网(有向网)当且仅当

1) $S \cup T \neq \emptyset$

到稿日期:2009-12-09 返修日期:2010-03-09 本文受国家自然科学基金(60673053, 90718011)资助。

吴振寰(1977—),男,硕士,讲师,主要研究方向为Petri网等,E-mail:wu_zhenhuan@163.com;高莹(1982—),女,硕士,助教,主要研究方向为Petri网等;吴哲辉(1941—),男,教授,主要研究方向为Petri网等。

- 2) $S \cap T = \emptyset$
- 3) $F \subseteq (S \times T) \cup (T \times S)$
- 4) $\text{dom}(F) \cup \text{cod}(F) = S \cup T$

其中

$$\text{dom}(F) = \{x \in S \cup T \mid \exists y \in S \cup T: (x, y) \in F\}$$

$$\text{cod}(F) = \{x \in S \cup T \mid \exists y \in S \cup T: (y, x) \in F\}$$

对 $x \in S \cup T$, 记

$$\cdot x = \{y \mid y \in S \cup T \wedge (y, x) \in F\}$$

$$x \cdot = \{y \mid y \in S \cup T \wedge (x, y) \in F\}$$

$\cdot x$ 和 $x \cdot$ 分别称作 x 的前集和后集, 并称 $\cdot x \cup x \cdot$ 为 x 的外延。

网 $N = (S, T; F)$ 称为一个简单网当且仅当

$$\forall x, y \in S \cup T, (\cdot x = \cdot y) \wedge (x \cdot = y \cdot) \rightarrow x = y$$

四元组 $\Sigma = (B, E; F, c)$ 称为一个基本网系统当且仅当

1) $(B, E; F)$ 是一个网, $c \subseteq B$ (B 称为条件集, E 称为事件集, c 称为 Σ 的一个情态)。

2) 事件 $e \in E$ 在情态 c 有发生权 (记作 $c[e]$) 当且仅当

$$\cdot e \subseteq c \wedge e \cdot \cap c = \emptyset$$

3) 若事件 e 在情态 c 发生, 产生新的情态 c' (记作 $c[e]$), 则

$$c' = (c - \cdot e) \cup e \cdot$$

如果一个基本网系统的网结构和情态满足更为严格的条件, 就可以构成 C/E 系统。

设 $\Sigma = (B, E; F, c_0)$ 为一个基本网系统, 其中 c_0 为初始情态, 如果 $C \subseteq \rho(B)$ 是满足条件

- 1) $c_0 \in C$;
- 2) $\forall c \in C, e_1, e_2 \in E, c_1, c_2 \subseteq B$
 $c_1[e_1]c \wedge c[e_2]c_2 \rightarrow c_1, c_2 \in C$

的最小集合, 则称 C 为 Σ 的一个完全情态集。

设 $\Sigma_1 = (B, E; F, c_0)$ 为一个基本网系统, C 是 Σ_1 的一个完全情态集。若 $(B, E; F)$ 是一个简单网, 而且

- 1) $\forall b \in B, \exists c_1, c_2 \in C$ 使得 $b \in c_1$ 但 $b \notin c_2$;
- 2) $\forall e \in E, \exists c_1, c_2 \in C$ 使得 $c_1[e]c_2$;

则称 $\Sigma = (B, E; F, C)$ 为一个条件/事件系统 (简称为 C/E 系统)。

在基本网系统 $\Sigma = (B, E; F, c)$ 中, 两个事件 $e_1, e_2 \in E$ 在情态 c 并发是指

- 1) $c[e_1] \wedge c[e_2]$;
- 2) $c[e_1]c_1 \rightarrow c_1[e_2] \wedge c[e_2]c_2 \rightarrow c_2[e_1]$ 。

两个事件在情态 c 并发的条件是

- 1) $(\cdot e_1 \cup e_1 \cdot) \cap (\cdot e_2 \cup e_2 \cdot) = \emptyset$;
- 2) $(\cdot e_1 \cup \cdot e_2) \subseteq c \wedge (e_1 \cdot \cup e_2 \cdot) \cap c = \emptyset$ 。

其中第 1) 个条件指出两个事件的外延不相交; 第 2) 个条件指出情态 c 同两个事件的外延之间的关系: 情态 c 必须覆盖两个事件的前集之并集, 而同两个事件的后集都不相交。

根据网论的观点, 若两个事件 e_1 和 e_2 并发, 则称它们为一个步 (并发步), 记为 $c[\begin{smallmatrix} e_1 \\ e_2 \end{smallmatrix}]$ 。这个记号强调了并发的非序性质。

在 CCS 中, 两个事件并发的条件其实是一样的, 只是处理方式不同。根据 CCS 的观点, 若事件 e_1 和 e_2 在情态 c 并发, 则它们可以按任意顺序发生, 即既可以是 $c[e_1 e_2]$, 也可以是 $c[e_2 e_1]$ 。可以看出, 这是列出并发事件作序列化处理的各可能出现的结果。

例 1 图 1 给出了一个基本网系统 $\Sigma_1 = (B, E; F, c_0)$ 。这个基本网系统中有 4 个事件和 5 个条件, 即 $B = \{b_1, b_2, b_3, b_4, b_5\}$, $E = \{e_1, e_2, e_3, e_4\}$, 系统的初始情态为 $c_0 = \{b_1, b_2\}$ 。

观察易知, 事件 e_1 和 e_2 在情态 c_0 并发, 即 $c_0[\begin{smallmatrix} e_1 \\ e_2 \end{smallmatrix}]$ 。根据 CCS 的观点, 在情态 c_0 有两个可能的发生序列, 即 $c_0[e_1 e_2]$ 和 $c_0[e_2 e_1]$ 。

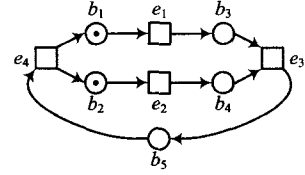


图 1 一个基本网系统 Σ_1

另一方面, 可以求出 Σ_1 的完全情态集为

$$C = \{\{b_1, b_2\}, \{b_2, b_3\}, \{b_1, b_4\}, \{b_3, b_4\}, \{b_5\}\}$$

容易验证 $\Sigma = (B, E; F, C)$ 是一个 C/E 系统。在这个系统中,

从情态 c_0 发生又回到 c_0 的极小步序列为 $\begin{pmatrix} e_1 \\ e_2 \end{pmatrix} e_3 e_4$ 。即

$$c_0[\begin{pmatrix} e_1 \\ e_2 \end{pmatrix} e_3 e_4] c_0$$

如果按照 CCS 的观点, 从 c_0 发生又回到 c_0 的极小事件序列 (我们称它们为“极小”, 是指该序列的任意非空子序列都不可能实现从 c_0 发生又回到 c_0 的功能) 有两个, 即

$$c_0[e_1 e_2 e_3 e_4] c_0 \text{ 和 } c_0[e_2 e_1 e_3 e_4] c_0$$

这是由于把 e_1 和 e_2 在 c_0 并发理解为可以按 $e_1 e_2$ 和 $e_2 e_1$ 两个顺序发生而引申出来的。我们把 $e_1 e_2 e_3 e_4$ 和 $e_2 e_1 e_3 e_4$ 称为交叠式序列 (由于 e_1 和 e_2 在 c_0 并发而产生的)。这是借助“交叠式并发”的名称而命名的。

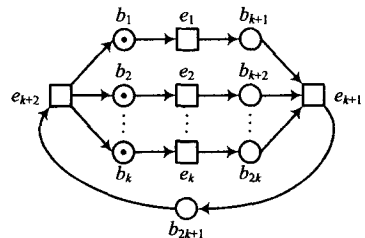
后面, 我们将以 C/E 系统为模型, 研究各种各样的并发情况下, 系统中交叠式序列的个数, 给出各种并发现象下, 系统的交叠式序列计算公式, 以此来进一步讨论“交叠式并发”与“真并发”两个概念的差异和联系。

3 C/E 系统中交叠式序列的数目计算

图 1 给出的是最简单的一种并发网系统。在这个网系统中, 只有两个事件在某情态并发。复杂的并发系统可能存在各种各样的并发情况, 例如, 多个事件在某情态下并发; 两个事件序列或多个事件序列的并发; 在一个网系统中存在多组互不相关的并发等。下面, 分别对各种并发情况下交叠式序列的数目计算进行讨论。

3.1 多个事件在某情态下并发的情况

这种情况的一般示意图如图 2 所示: 有 k ($k \geq 2$) 个事件 $e_1, e_2, \dots, e_k$ 在情态 $c_0 = \{b_1, b_2, \dots, b_k\}$ 并发。我们用 $C(k)$ 表示这种并发情况。



根据 CCS 的理解,事件 $e_1, e_2, \dots, e_k$ 可以按任意顺序发生。由于 k 个事件的不同排列有 $k!$ 种,因此,在这种系统中,交叠式序列的个数有 $k!$ 个。我们用符号 $NAS()$ 表示并发系统中交叠式序列的数目,其中括号内注明系统中并发的情况(种类),就可以归纳出下面定理。

定理 1 设 $\Sigma_1 = (B, E; F, c_0)$ 为一个基本网系统; C 为 Σ_1 的完全情态集; $\Sigma = (B, E; F, C)$ 为一个条件事件系统。若 Σ_1 中有(且仅有一组) k 个事件在某情态并发,则

$$NAS(C(k)) = k! \quad (1)$$

例 2 图 3 是一个有 3 个事件(e_1, e_2 和 e_3)在某情态(c_0)处于并发的基本网系统。容易验证,这个基本网系统的完全情态集同基本网结构一起构成 C/E 系统。在这个网系统中,交叠式序列共有 $3! = 6$ 个。它们分别为

$$\sigma_1: e_1 e_2 e_3 e_4 e_5, \quad \sigma_2: e_1 e_3 e_2 e_4 e_5, \quad \sigma_3: e_2 e_3 e_1 e_4 e_5$$

$$\sigma_4: e_2 e_1 e_3 e_4 e_5, \quad \sigma_5: e_3 e_1 e_2 e_4 e_5, \quad \sigma_6: e_3 e_2 e_1 e_4 e_5$$

它们都是满足 $c_0[\sigma_i]c_0 (i \in \{1, 2, 3, 4, 5, 6\})$ 的最短序列。

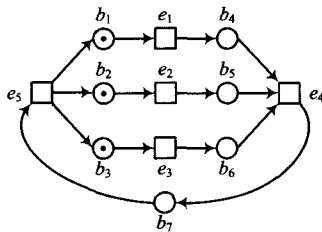


图 3 有 3 个事件在情态 c_0 并发的基本网系统

3.2 多个事件序列在某情态下并发的情况

我们先讨论两个事件序列在某情态并发的情况。图 4 所示是一个有两个事件序列并发的基本网系统,其中 $c_0 = \{b_{11}, b_{12}\}$ 。这个基本网系统的完全情态集同网结构也构成 C/E 系统。观察易知,在这个网系统中,事件 e_{11} 和 e_{12} 在初始情态 c_0 并发。如果 e_{11} 在初始情态 c_0 发生,产生新的情态为 $c_1 = \{b_{12}, b_{21}\}$,即 $c_0[e_{11}]c_1$,则事件对 e_{12} 和 e_{21} 在 c_1 并发。反之,如果 e_{21} 在 c_0 发生,产生新的情态为 $c_2 = \{b_{11}, b_{22}\}$,即 $c_0[e_{21}]c_2$,则事件对 e_{11} 和 e_{22} 在 c_2 并发。……这样的情况一直延续下去,直到事件对 e_{1l_1} 和 e_{2l_2} 在情态 $\{b_{1l_1}, b_{2l_2}\}$ 并发为止。我们把这种情况称为事件序列 $e_{11} e_{12} \dots e_{1l_1}$ 和事件序列 $e_{21} e_{22} \dots e_{2l_2}$ 在情态 c_0 并发。事件 e_3 是这两个并发的事件序列的同步事件。对于这种两个长度(即事件数)分别为 l_1 和 l_2 的事件序列在某情态并发的情况,用符号 $C(l_1, l_2)$ 表示。

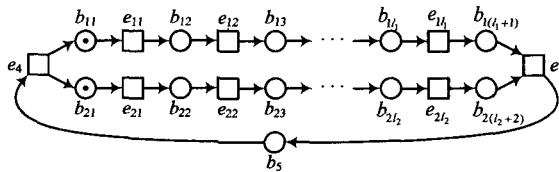


图 4 两个事件序列在某情态并发的基本网系统

下面讨论这种网系统中交叠式序列的计算。我们对 l_2 的取值从 1 开始进行归纳。

易知,当 $l_2 = 1$,即第 2 个事件序列只含一个事件 e_2 时,这个系统的交叠式序列共有 $l_1 + 1$ 个,因为 e_2 可以插入序列 $e_{11} e_{12} e_{13} \dots e_{1l_1}$ 的任何一个位置:序列的最前面、最后面或序列中任意两个事件之间。因此有

$$NAS(C(l_1, 1)) = l_1 + 1 \quad (2)$$

当 $l_2 = 2$ 时,设第 2 个序列为 $e_{21} e_{22}$ 。我们先考察把 e_{21} 插

入序列 $e_{11} e_{12} \dots e_{1l_1}$ 中。从前面的讨论知,共有 $l_1 + 1$ 个插入位置。 e_{21} 插入以后,把 e_{22} 插入由 $e_{11} e_{12} \dots e_{1l_1}$ 和 e_{21} 合成的序列时,由于 e_{22} 在序列中的位置必须在 e_{21} 之后,因此对于 e_{21} 不同的插入情况, e_{22} 可以插入的位置个数是不同的。譬如,在序列 $e_{21} e_{11} e_{12} \dots e_{1l_1}$ 中, e_{22} 有 $l_1 + 1$ 个插入位置;在序列 $e_{11} e_{21} e_{12} \dots e_{1l_1}$ 中, e_{22} 有 l_1 个插入位置;而在序列 $e_{11} e_{12} \dots e_{1l_1} e_{21}$ 中, e_{22} 只有一个插入位置。可见,当 $l_2 = 2$ 时,交叠式序列的个数为

$$(l_1 + 1) + l_1 + \dots + 2 + 1 = \sum_{i=1}^{l_1+1} i \quad (3)$$

注意,对任意 $i \geq 2$ 都有

$$i = NAS(C(i-1, 1))$$

如果记

$$NAS(C(0, 1)) = 1$$

那么就有

$$NAS(C(l_1, 2)) = \sum_{i=1}^{l_1+1} NAS(C(i-1, 1)) = \sum_{i=0}^{l_1} NAS(C(i, 1))$$

以上的讨论可以推广到一般情况。当第 2 个序列有 l_2 个事件时,可以假设前面 $l_2 - 1$ 个元素都插入以后,再考察 e_{l_2} 插入。从而易知

$$NAS(C(l_1, l_2)) = \sum_{i=0}^{l_1} NAS(C(i, l_2 - 1))$$

可把上面的讨论归结为如下的定理。

定理 2 设 $\Sigma_1 = (B, E; F, c_0)$ 为一个基本网系统; C 是 Σ_1 的完全情态集; $\Sigma = (B, E; F, C)$ 构成一个 C/E 系统。如果 Σ_1 中有(且仅有一组)长度分别为 l_1 和 l_2 的两个事件序列并发,那么该网系统中交叠式序列的个数可通过下面的递推公式计算。

$$\begin{cases} NAS(C(0, i)) = 1, & i \geq 1 \\ NAS(C(i, 1)) = i + 1, & i \geq 1 \\ NAS(C(l_1, l_2)) = \sum_{i=0}^{l_1} NAS(C(i, l_2 - 1)) \end{cases}$$

下面讨论多于两个事件序列并发的情况。设并发的事件序列有 $k (k \geq 2)$ 个,各个序列的长度分别为 $l_1, l_2, \dots, l_k$ 。我们把问题理解为在前 $k - 1$ 个并发序列形成的各个部分交叠式序列的基础上,如何把第 k 个序列中的各个事件插入到各个部分交叠式序列中。从而可以得到下面的推论。

推论 设 $\Sigma_1 = (B, E; F, c_0)$ 为一个基本网系统; C 是 Σ_1 的完全情态集; $\Sigma = (B, E; F, C)$ 构成一个条件系统。如果 Σ_1 中有(且仅有一组)长度分别为 $l_1, l_2, \dots, l_k$ 的 $k (k \geq 2)$ 个时间序列并发,那么该网系统中交叠式序列的个数可以通过下面递推公式计算。

$$\begin{cases} NAS(C(0, i)) = 1, & i \geq 1 \\ NAS(C(i, 1)) = i + 1, & i \geq 1 \\ NAS(C(l_1, l_2, \dots, l_k)) = \sum_{i=0}^{NAS(C(l_1, l_2, \dots, l_{k-1})) - 1} NAS(i, l_k - 1) \end{cases}$$

例 3 图 5 所示的基本网系统是一个有 3 个事件序列在 c_0 并发的系统,这 3 个事件序列分别为 $e_{11} e_{12} e_{13} e_{14}$, $e_{21} e_{22}$ 和 $e_{31} e_{32} e_{33}$ 。

为了计算这个网系统中交叠式序列的数目,我们可以先计算由第一个事件序列 $e_{11} e_{12} e_{13} e_{14}$ 同第 2 个事件序列 $e_{21} e_{22}$ 合成的部分交叠式序列的个数。即

$$NAS(C(4,2)) = \sum_{i=0}^4 C(i,1) = 5+4+3+2+1=15$$

然后再考虑把第3个事件序列 $e_{31} e_{32} e_{33}$ 插入上述所得到的15个部分交叠式序列中。这样就有

$$\begin{aligned} NAS(C(4,2,3)) &= \sum_{i=0}^{NAS(C(4,2))-1} NAS(C(i,2)) \\ &= \sum_{i=0}^{14} NAS(C(i,2)) = \sum_{j=1}^{15} \sum_{i=1}^j i = 480 \end{aligned}$$

也就是说,根据 CCS 对并发的理解,这个并发系统共有 480 个交叠式序列。这是一个不小的数目。

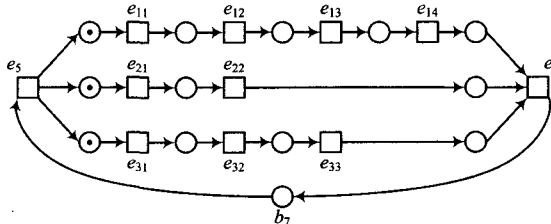


图5 一个有3个事件序列并发的基本网系统

3.3 存在多个互不相关的并发事件组的情况

我们把 3.1 节所述的多个事件并发和 3.2 节所述的多个事件序列并发的情况中全体事件或事件序列的集合作为一个并发组。在一个基本网系统中若有两个并发组,分别记为 C_A 和 C_B ,如果 C_A 中的任一个事件和 C_B 中的任一个事件在每个可达状态下都不处于并发关系,就说 C_A 和 C_B 是两个互不相关的并发组。图 6 和图 7 的基本网系统都有两个互不相关的并发组。并发组 C_A 是一个事件组 $\{e_1, e_2, e_3\}$,并发组 C_B 是一个事件序列组 $\{e_{41} e_{42}, e_{51} e_{52} e_{53}\}$ 。在图 6 的网系统中, C_A 和 C_B 是一种顺序关系,记为 $C_A \circ C_B$;在图 7 的网系统中 C_A 和 C_B 是一种选择关系,记为 $C_A + C_B$ 。

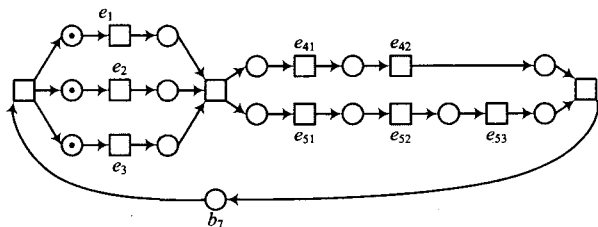


图6 有两个互不相关的并发组的基本网系统(两个并发组存在顺序关系)

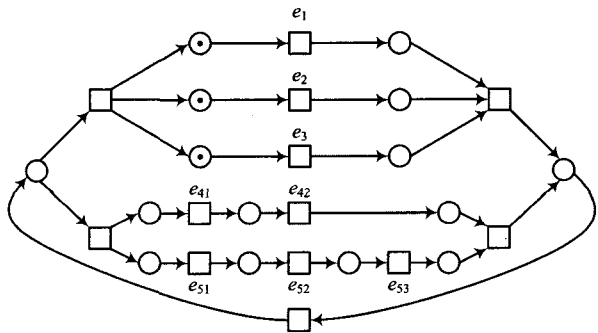


图7 有两个互不相关的并发组的基本网系统(两个并发组存在选择关系)

易知,在基本网系统中的两个互不相关的并发组存在顺序关系时,整个网系统的交叠式序列的个数是两个并发组各自产生的交叠式序列个数的乘积;而当两个互不相关的并发组存在选择关系时,整个网系统的交叠式序列的个数是它们

各自产生的交叠式序列的个数之和。我们把这些结论归纳为下面的定理。

定理 3 设 $\Sigma_1 = (B, E; F, C_0)$ 为一个基本网系统; C 是 Σ_1 的完全情态集; $\Sigma = (B, E; F, C)$ 构成一个 C/E 系统。若 Σ_1 中有两个互不相关的并发组 C_A 和 C_B ,那么

$$NAS(C_A \circ C_B) = NAS(C_A) \cdot NAS(C_B)$$

$$NAS(C_A + C_B) = NAS(C_A) + NAS(C_B)$$

例 4 对于图 6 的基本网系统,可以求出它的交叠式序列的个数为

$$NAS(C(3) \circ C(2,3)) = NAS(C(3)) \cdot NAS(C(2,3))$$

其中

$$NAS(C(3)) = 3! = 6$$

$$NAS(C(2,3)) = \sum_{i=0}^2 NAS(C(i,2)) = 1+3+6=10$$

从而

$$NAS(C(3) \circ C(2,3)) = 6 \cdot 10 = 60$$

图 7 所示的基本网系统的交叠式序列个数则为

$$\begin{aligned} NAS(C(3) + C(2,3)) &= NAS(C(3)) + NAS(C(2,3)) \\ &= 6 + 10 = 16 \end{aligned}$$

当一个基本网系统中存在 3 个甚至更多的互不相关的并发组时,也可以根据它们之间的联接关系写出交叠式序列个数的计算公式。

例如,图 8 和图 9 都是存在 3 个互不相关的并发组的基本网系统的示意图。根据这些并发组的联接关系易知:

$$NAS(C_A \circ C_B + C_C) = NAS(C_A) \cdot NAS(C_B) + NAS(C_C)$$

$$NAS(C_A \circ (C_B + C_C)) = NAS(C_A) \cdot (NAS(C_B) + NAS(C_C))$$

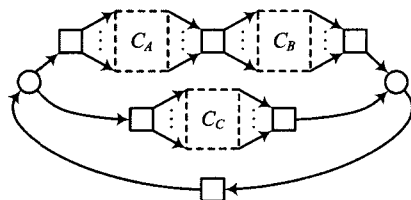


图8 存在3个互不相关的并发组的一个基本网系统(示意图)

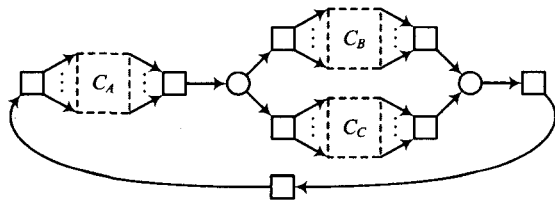


图9 存在3个互不相关的并发组的另一个基本网系统(示意图)

结束语 本文以 C/E 系统为模型,研究了 Petri 网和 CCS 中关于并发的定义之间的差异和联系。CCS 把两个事件并发定义为它们可以按任意顺序发生,这个关于并发的定义被称为“交叠式并发”;而 Petri 网的并发定义主要强调其非序性质,这种定义被称为“真并发”。

可以认为,交叠式并发实质上是对并发事件进行序列化处理的一种方式,或者说,它指出了对并发事件作序列化处理

(下转第 229 页)

推论 1 设 $\varphi \in \text{TITL}_R$, M 为模型空间, 则 $\exists (m \in M, m \models \varphi)$ 不可判定。

推论 2 设 $\varphi \in \text{TITL}_R$, $A \in \text{TA}_R$, 则 $A \models \varphi$ 不可判定。
由定理 1 和定理 2, 易得以下推论。

推论 3 $L_{\text{TITL}_R} \cap L_{\text{TA}_R} = L_{\text{TITL}_{R-\text{SUB}}}$

引理 6^[2] $L_{\text{TA}_R} \supseteq L_{\text{MITL}}$

推论 4 $L_{\text{TITL}_R} \Leftrightarrow L_{\text{TITL}_{R-\text{SUB}}} \supseteq L_{\text{MITL}}$

结束语 我们的结果是定理 3、定理 4、推论 1、推论 2、推论 3、推论 4。这些定理从 3 方面给出了结论。

关于 TITL 满足性判定问题。定理 3 证明了 $\text{TITL}_{R-\text{SUB}}$ 满足性可判定; 定理 4 证明了 $\text{TITL}_{R-\text{SUB}}$ 性质可模型检测; 而推论 1、推论 2 则证明, TITL_R 满足性不可判定, 也不可模型检测。这些定理表明: 不允许在“逻辑非”的辖域内出现时间约束公式, 这样得到的实数域时间区间时序逻辑的约束子集所描述的实时系统 TITL 性质, 是可以开发出工具来自动验证的。而“非”辖域内出现时间约束这样的性质, 则不可自动验证。

关于 TITL 可判定子集的判定效率问题。定理 3 和定理 4 表明: $\text{TITL}_{R-\text{SUB}}$ 判定可满足性与模型检测的问题固有时间复杂度的下限是非初等。也就是说, 任何相应的实时系统模型检测工具在最坏情况下, 至少需要非初等时间来判定模型是否满足给定的 TITL 性质。

关于 TITL 逻辑表达能力问题。推论 3 和推论 4 表明: 时间区间时序逻辑 TITL_R 的表达能力超出了时间自动机所接受的时间正则语言的范围, 如图 1 所示, 与表达能力低于时间自动机的普通实时逻辑 MITL^[2,4] 相比, 区间实时逻辑无疑有更强大的描述工业实时系统性质的能力。

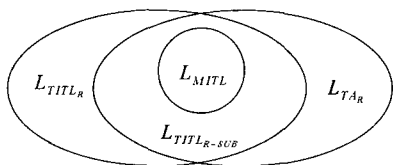


图 1 几种逻辑与自动机的表达能力

参考文献

- [1] Alur R, Dill D L. A Theory of Timed Automata[J]. Theoretical Computer Science, 1994, 126(2): 183-236
- [2] Alur R, Feder T, Henzinger T A. The benefits of relaxing punct-

uality[J]. Journal of the ACM, 1996, 43(1): 116-146

- [3] Alur R, Henzinger T A. A really temporal logic[J]. Journal of the ACM, 1994, 41(1): 181-204
- [4] Alur R, Henzinger T A. Logics and models of real time; A survey[C]//Lecture Notes in Computer Science. Springer-Verlag, 1992, 600: 74-106
- [5] Thomas A, Henzinger T A, Manna Z, et al. What good are digital clocks?[C]//Proc. ICALP'92, volume 623 of LNCS. Springer, 1992: 545-558
- [6] Zhou C, Hoare C A, Ravn A P. A calculus of duration[J]. Information Processing Letters, 1991, 40(5): 269-276
- [7] Li G, Tang Z. Translating a Continuous-Time Temporal Logic into Timed Automata[C]//Proceedings of the first Asian Symposium on Programming Languages and Systems (APLAS 2003). Lecture Notes in Computer Science 2895. Springer-Verlag, 2003: 322-338
- [8] Duan Z. Modeling of hybrid systems[M]. Beijing: science press, 2004: 11-30
- [9] Duan Z, Koutny M. A framed temporal logic programming language[J]. Journal of Computer Science and Technology, 2004, 19(3): 314-351
- [10] Wilke T. Specifying timed state sequences in powerful decidable logics and timed automata[C]//Lecture Notes in Computer Science. Springer, 1994, 863: 694-715
- [11] Tian C, Duan Z. Propositional Projection Temporal Logic, Büchi Automata and omega-Regular Expressions[C]//Proceedings, Theory and Applications of Models of Computation, Lecture Notes in Computer Science, 4978. Springer, 2008: 47-58
- [12] Moszkowski B. Reasoning about Digital Circuits[D]. Stanford University, 1983
- [13] Fränzle M. Model-checking dense-time duration calculus [J]. Formal Aspects of Computing, 2004, 16(2): 121-139
- [14] Zhou C, Hansen M R, Sestoft P. Decidability and Undecidability Results for Duration Calculus[C]//STACS'93, Lecture Notes in Computer Science, 665. Springer-Verlag, 1993: 58-68
- [15] Duan Z, Tian C, Zhang L. A decision procedure for propositional projection temporal logic with infinite models[J]. Acta Informatica, 2008, 45(1): 43-78
- [16] 张海宾, 段振华. 混合投影时序逻辑与混合系统的形式化验证[J]. 计算机科学, 2007, 34(11): 279-282

(上接第 226 页)

时存在的各种可能性。因此, 本文把并发循环网系统(如 C/E 系统)中对并发事件序列化处理后产生的循环序列称之为交叠式序列。进一步, 我们讨论了各种不同的并发情况下并发循环网系统中交叠式序列数目的计算, 给出了相应的计算公式。通过这些计算, 可以进一步指示两种不同的并发概念之间的差异和联系。

参考文献

- [1] 吴鹤龄, 崔林. ACM 图灵奖(1966-2006)—计算机发展史的缩影(2 版)[M]. 北京: 高等教育出版社, 2008
- [2] Milner R. Flow graph and flow algebras[J]. Journal of Association for Computing Machinery, 1979, 26(4): 794-818

- [3] Milner R. A calculus of communicating system [J]. Lecture Notes in Computer Science, 1980, 92: 1-168
- [4] Petri C A. Kommunikation mit automaten[J]. Bonn: Institue fur Instrumentelle Mathematik, Suhrifon des IIM Nr. 2, 1962 (In German)
- [5] 袁崇义. Petri 网[M]. 南京: 东南大学出版社, 1989
- [6] Petri C A. Concurrency[J]. Lecture Notes in Computer Science, 1980, 84: 251-260
- [7] Genrich H J, Lautenback K, Thiagrajan P S. Elements of general net theory[J]. Lecture Notes in Computer Science, 1980, 84: 21-163
- [8] 袁崇义. Petri 网原理与应用[M]. 北京: 电子工业出版社, 2005
- [9] 吴哲辉. Petri 网导论[M]. 北京: 机械工业出版社, 2006